IN THE CLAIMS:

1. (currently amended) A context controller for managing multitasking in a processor, comprising:

an event recorder that records occurrences of predetermined events; and

an event acknowledger, associated with said event recorder, that acknowledges ones of said events based on code of ~~which are relevant to~~ a currently-active context.

2. (original) The context controller as recited in Claim 1 further comprising an event masker, associated with said event recorder, said event acknowledger and each context executing on said processor, that masks others of said events as a function of said each context.

3. (original) The context controller as recited in Claim 1 wherein said event recorder is embodied in at least one flip-flop within said context controller.

4. (original) The context controller as recited in Claim 1 further comprising;

a foreground task controller that activates contexts corresponding to foreground tasks based on priority and in response to said events; and

a background task controller that cyclicly activates contexts corresponding to said background tasks subject to activation of said contexts corresponding to said foreground tasks.

5. (currently amended) The context controller as recited in Claim 1 further comprising a background task controller that activates contexts corresponding to background tasks based on numbers of instructions executed by each of said background tasks, wherein said each of said background tasks accomplishes an equal amount of work before a cycle of background processing repeats.

6. (original) The context controller as recited in Claim 1 wherein said context controller places said processor in an idle state when all foreground and background tasks are inactive.

7. (original) The context controller as recited in Claim 1 further comprising a foreground task controller adapted to activate a context corresponding to a particular foreground task by vectoring to a software-selectable memory location.

8. (currently amended) A method of managing multitasking in a processor, comprising the steps of:

recording occurrences of predetermined events; and

acknowledging ones of said events based on code of ~~which are relevant to~~ a currently-active context.

9. (original) The method as recited in Claim 8 further comprising the step of masking others of said events as a function of each context executing on said processor.

10. (original) The method as recited in Claim 8 wherein said step of recording comprises the step of changing a state of at least one flip-flop within said context controller.

11. (original) The method as recited in Claim 8 further comprising the steps of:

activating contexts corresponding to foreground tasks based on priority and in response to said events; and

cyclicly activating contexts corresponding to said background tasks subject to activation of said contexts corresponding to said foreground tasks.

12. (currently amended) The method as recited in Claim 8 further comprising the step of activating contexts corresponding to background tasks based on numbers of instructions executed

by each of said background tasks, wherein said each of said background tasks accomplishes an equal amount of work before a cycle of background processing repeats.

13. (original) The method as recited in Claim 8 further comprising the step of placing said processor in an idle state when all foreground and background tasks are inactive.

14. (original) The method as recited in Claim 8 further comprising the step of activating a context corresponding to a particular foreground task by vectoring to a software-selectable

15. (currently amended) A processor, comprising:

an instruction decoder that decodes instructions received into said processor and corresponding to a plurality of tasks;

a plurality of register sets, corresponding to said plurality of tasks, that contain operands to be manipulated;

an execution core, coupled to said instruction decoder and said plurality of register sets, that executes instructions corresponding to an active one of said plurality of tasks to manipulate ones of said operands; and

a context controller for managing multitasking in said processor, including:

an event recorder that records occurrences of predetermined events; and

an event acknowledger, associated with said event recorder, that acknowledges ones of said events based on code of a ~~which are relevant to~~ currently-active context.

16. (original) The processor as recited in Claim 15 wherein said context controller further includes an event masker, associated with said event recorder, said event acknowledger and each context executing on said processor, that masks others of said events as a function of said each context.

17. (original) The processor as recited in Claim 15 wherein said event recorder is embodied in at least one flip-flop within said context controller.

18. (original) The processor as recited in Claim 15 wherein said context controller further includes:

a foreground task controller that activates contexts corresponding to foreground tasks based on priority and in response to said events; and

a background task controller that cyclicly activates contexts corresponding to said background tasks subject to activation of said contexts corresponding to said foreground tasks.

19. (currently amended) The processor as recited in Claim 15 wherein said context controller further includes a background task controller that activates contexts corresponding to background tasks based on numbers of instructions executed by each of said background tasks, wherein said each of said background tasks accomplishes an equal amount of work before a cycle of background processing repeats.

20. (original) The processor as recited in Claim 15 wherein said context controller places said processor in an idle state when all foreground and background tasks are inactive.

21. (original) The processor as recited in Claim 15 wherein said context controller further includes a foreground task controller adapted to activate a context corresponding to a particular foreground task by vectoring to a software-selectable memory location.

22. (original) The processor as recited in Claim 15 wherein said processor forms a portion of a general-purpose computer.